**1 招标者只向一些经过筛选合格的供应商发出投标邀请，这种招标方式是（）**

这种招标方式被称为邀请招标或有限制的招标。它与公开招标不同，在邀请招标中，招标人只向特定的供应商发出邀请，而在公开招标中，任何有资格的供应商都可以参加。

**2 创建一个项目的WBS，可以（）**

1. 首先，明确项目的目标和范围，并将其分解为较小、更易管理的工作包。

2. 将这些工作包进一步分解为更小的任务和活动，并将它们组织成一个层次结构，每个级别表示逐渐细化的范围。

3. 确定每个任务和活动的持续时间、资源需求和交付要求等详细信息，并将它们记录在一个项目计划中。

4. 对于每个任务和活动，指定一个负责人，并确定他们的职责和任务完成的期限。

5. 定期监测项目进度和绩效，并对WBS进行必要的调整和更新，以确保项目按时交付并达到预期目标。

**3 对某个项目活动的持续时间进行三点估算，得到其最乐观时间为 8 天，最悲观时间为 24 天，最可能时间为 10 天，则该活动的持续时间期望值是（）怎样计算？**

该活动的持续时间期望值可以通过三点估算法求得，其计算公式如下：

(Estimated Duration) = (Optimistic Time + 4 x Most Likely Time + Pessimistic Time) / 6

将最乐观时间为 8 天，最悲观时间为 24 天，最可能时间为 10 天代入公式中得：

(Estimated Duration) = (8 + 4 x 10 + 24) / 6 = 14 天

**4 快速跟进是指（）** 在项目管理中，对项目关键路径上的任务和活动进行重点跟进和控制，以确保项目能够按时交付和达到预期目标。快速跟进通常与敏捷方法和灵活的项目管理实践有关，它强调了快速反应、紧密协作和迭代开发的重要性，以便在项目进展过程中及时调整和优化计划，从而更好地满足客户需求并提高项目绩效。

**5 （）用来度量软件规模，与实现产品所使用的语言没有关系。**

功能点是用来度量软件规模的方法，与实现产品所使用的语言没有关系。功能点主要考虑产品能够提供的功能和交互，而不涉及具体的技术实现。这种方法可以帮助软件开发团队更好地估算项目的规模、成本和时间，并为客户提供明确的交付标准和质量保证。常见的功能点计算方法包括IFPUG（International Function Point Users Group）和COSMIC（Common Software Measurement International Consortium）等。

**6 在基本 COCOMO 模型中，用一个以（）为自变量的函数来计算软件开发工作量。**

在基本COCOMO模型中，用一个以代码行数为自变量的函数来计算软件开发工作量。具体来说，这个函数的形式是：Effort = a(KLOC)^b 其中，Effort表示软件开发的工作量，KLOC表示代码行数（以千行为单位），a和b是经验常数，代表了不同的项目类型、技术要求和团队能力对工作量的影响。该模型假设软件规模与工作量之间存在着指数关系，即随着代码行数增加，工作量也会相应增加，并通过使用历史数据来确定适当的系数。

**7 与其他软件系统相互交换信息的能力称为软件的（）**

与其他软件系统相互交换信息的能力称为软件的互操作性（interoperability）。互操作性是指不同的软件系统之间能够进行有效的交互和数据共享，以实现更高效的业务流程、增强软件应用的价值和可扩展性，并减少不必要的工作和错误。互操作性一般包括协议、接口、标准和格式等方面的内容，可以通过开放式架构、API和协同开发等手段来实现。在当前的信息化时代中，互操作性已经成为了许多软件系统设计和实现的重要目标之一。

**8 未调整功能点数 UFC 的功能计数项（）**

未调整功能点数（Unadjusted Function Points，UFP）是指在计算软件规模时，根据用户需求文档和功能清单，对软件的功能进行分类、量化和评估，得到的原始功能点数。未调整功能点数主要用于计算软件开发过程中的工作量和生产率，并作为评价软件质量和复杂度的重要指标之一。 未调整功能点数包括以下计数项：

1. 外部输入：由用户输入的数据或信息，如在线表单、报告、查询等。

2. 外部输出：向用户输出的数据或信息，如网页、报告、图表等。

3. 外部查询：用户通过系统查询所获得的信息，如搜索结果、查询列表等。

4. 内部逻辑文件：系统内部存储的数据或信息，如数据库、文件、消息等。

5. 外部接口文件：与其他系统交互所使用的数据或信息，如API、XML、JSON等。

以上计数项都是未调整功能点数中的组成部分，通过对它们进行量化和评估，可以得到一个软件项目的原始功能点数。

**判断题 1 间接成本是与一个具体的项目相关的成本（错）。**

判断：间接成本不是与一个具体的项目相关的成本。 间接成本是指在生产或营销过程中不能直接归属于某个特定产品或服务的支出，如管理费用、办公室租金、水电费等。这些成本通常无法直接追踪到单个项目，而是通过分摊方法对多个项目进行分配。因此，间接成本是与公司整体运营相关的成本，而非与某个具体项目相关。

**2 判断：在软件项目中，产品范围就是项目范围。（错）**

判断：在软件项目中，产品范围不一定就是项目范围。 产品范围和项目范围是两个不同的概念。产品范围指的是要开发的软件产品的功能、性能、特点等方面的要求和限制；而项目范围包括了完成项目所需开展的所有活动、交付物和里程碑等。显然，产品范围只是项目范围的一个组成部分，而并不代表整个项目的范围。在实际项目中，这两者之间可能存在重叠，但也有可能会出现差异。

**3 判断：范围控制要通过变更控制系统和配置管理系统来完成。（错）**

判断：范围控制要通过变更控制系统来完成，但不一定需要配置管理系统。范围控制是指在整个项目生命周期中，对于项目范围的识别、定义、确认和变更进行有效的管理和控制。其中，变更控制系统是实现范围控制的重要手段之一，它可以帮助项目团队监测项目的执行情况，及时识别并处理范围变更请求，确保只有经过评估和批准的变更才能执行。而配置管理系统则是一种多功能管理工具，在软件开发等领域使用较为普遍，其主要作用是记录和跟踪软件配置项（如代码、文档等）的变更历史和版本信息，以便于管理和维护。虽然在某些项目中可能也会使用配置管理系统来辅助范围控制工作，但它并不是必需的，因为范围控制主要关注的是项目的范围变更管理，而非特定的配置项变更管理。

**4 判断：技术评审可以在软件项目的任何阶段执行，因此可以尽早发现和消除缺陷。（错）**

判断：技术评审不可以在软件项目的任何阶段执行，应该在特定的时间节点进行。技术评审是一种通过专业人员对软件项目或产品进行全面、系统的检查和审查，以发现和消除缺陷的方法。尽管技术评审可以帮助提高软件品质，但它并不是可以在任何时间执行的活动，而应该在特定的时间节点进行。具体来说，技术评审通常分为以下几个阶段：

1. 设计评审（Design Review）：在软件设计完成后，对其进行检查和审查，以确保设计满足要求、符合规范和标准，并且具有可行性和实现性。

2. 代码评审（Code Review）：在编写代码的过程中，对代码进行检查和审查，以确保代码的正确性、可读性、可维护性和可测试性。

3. 集成评审（Integration Review）：在软件各模块集成完成后，对整体进行检查和审查，以确保各模块间的接口和交互正常，并且整体满足要求。

4. 测试评审（Test Review）：在软件测试完成后，对测试结果进行检查和审查，以确保测试覆盖了所有需求，并且测试用例设计合理、测试执行正确、测试报告完整。

因此，技术评审应该在特定的时间节点进行，以便及时发现和消除缺陷，提高软件品质。

**5 判断：代码评审可以比测试更为有效地发现一些类型的程序逻辑错误。（错）**

判断：代码评审可以在一定程度上发现一些类型的程序逻辑错误，但测试仍然是发现程序逻辑错误最为有效的方法之一。代码评审和测试是常用的软件工程技术，它们都可以帮助发现并消除程序中的缺陷。相比于测试，代码评审主要关注的是代码质量、可读性和可维护性等方面，能够帮助检查代码是否符合编码规范、代码风格是否一致、变量命名是否合理等问题，从而提高代码的质量和可维护性。另外，在代码评审过程中，也有可能发现一些基本的程序逻辑错误（如语法错误、拼写错误等），但相对于复杂的程序逻辑错误，代码评审的效果还有限。

与之相比，测试则更加注重功能完整性、正确性和可靠性等方面，能够帮助发现各种类型的程序逻辑错误，包括边界问题、数据结构错误、算法错误、并发问题等。因此，测试仍然是发现程序逻辑错误最为有效的方法之一，在软件开发生命周期中占据着重要地位。

**简答题 1 什么是项目活动的最早和最迟开始时间、最早和最迟结束时间？什么是项目活动的总浮动时间和自由浮动时间？**

项目活动的最早和最迟开始时间（Early Start，ES 和 Late Start，LS）是指在网络图中，该活动能够开始执行的最早和最晚时间。最早开始时间是指在不影响后续活动的前提下，该活动能够尽早地开始执行所需的时间；最迟开始时间则是指该活动必须在不推迟整个项目进度的前提下，最晚可以延迟到哪个时间点才开始执行。

同样地，项目活动的最早和最迟结束时间（Early Finish，EF和Late Finish，LF）是指在网络图中，该活动能够完成执行的最早和最晚时间。最早结束时间是指在不影响后续活动的前提下，该活动能够尽早地完成所需的时间；最迟结束时间则是指该活动必须在不推迟整个项目进度的前提下，最晚可以延迟到哪个时间点才能完成。

总浮动时间（Total Float，TF）是指一个项目活动可以自由延迟的时间，而不会对项目完成时间造成影响的时间差。它等于该活动的最晚结束时间减去最早结束时间。总浮动时间越多，表示该活动的弹性越大，对项目进度的风险影响也就越小。

自由浮动时间（Free Float，FF）是指在不影响后续活动的前提下，一个项目活动可以自由延迟的时间。它等于该活动的后继活动的最早开始时间减去该活动的最早结束时间。自由浮动时间越多，表示该活动对后续活动的影响越小，可以更灵活地调整活动的执行顺序，从而提高项目的弹性和可控性。

**2 什么是软件项目的规模、工作量和成本？它们一般用什么度量单位来度量？**

软件项目的规模、工作量和成本是项目管理中常用的三个指标，它们分别用来评估和度量软件项目的大小、复杂度和投入。

1. 规模（Size）：软件项目的规模通常指开发或交付的软件产品所包含的代码行数、功能点、用例数等。规模越大，代表项目的复杂度和难度也就越高。

2. 工作量（Effort）：软件项目的工作量通常指完成整个项目所需要的人力资源，包括开发、测试、管理等方面的人力投入。工作量是衡量软件项目成本的重要因素之一。

3. 成本（Cost）：软件项目的成本通常指完成整个项目所需的经济支出，包括人力资源、设备、软件工具、培训、运营等各方面的费用。成本是评估软件项目经济效益的重要指标。

这三个指标一般会使用不同的度量单位来进行度量：

1. 规模可以使用代码行数、功能点、用例数等来进行度量，其中最为常用的是功能点。功能点是一种通过对软件功能进行分类和计算来度量软件规模的方法，常用的功能点计算方法包括 IFPUG和 COSMIC。

2. 工作量通常使用人天、人月、人年等来进行度量。其中，人天是指一个人工作一天所需的时间量，而人月则是指一个人在一个月内工作所需的时间量。另外，为了更精确地评估工作量，也可以使用软件工程中的其他指标来辅助计算，如功能点、代码行数等。

3. 成本通常使用货币单位（如美元、人民币等）来进行度量。常用的成本计算方法包括费用估算、成本效益分析、投资回报率等。此外，也可以使用其他指标来计算成本，如人均效率、生产力等。

**计算题 1 某项目原来预计到 2015 年 10 月 10 日完成 10 万元的工作，但是到该日期时只完成了其中 8.5 万元的工作，而为了完成这些工作实际花费了 9 万元。请用挣值分析法计算在 2015 年 10 月 10 日项目的成本偏差、进度偏差、成本效能指数和进度效能指数各是多少？**

挣值分析法是一种常用的项目成本管理技术，通过对项目的实际工作量和花费情况进行统计和比较，计算出项目的成本偏差、进度偏差、成本效能指数和进度效能指数等指标，从而帮助评估项目的绩效和风险。

在本题中，我们可以按照以下步骤来计算各项指标：

1. 计算预算成本（Budgeted Cost of Work Scheduled，BCWS）和实际成本（Actual Cost of Work Performed，ACWP）。

预算成本是指在预算计划中规定的截止日期前计划完成的工作应该花费的成本。根据题目描述，该项目预计到 2015 年 10 月 10 日完成 10 万元的工作，因此其预算成本 BCWS 为 10 万元。

实际成本是指到截至日期为止实际发生的成本。根据题目描述，实际完成了其中 8.5 万元的工作，但实际花费了 9 万元，因此其实际成本 ACWP 为 9 万元。

2. 计算实际进度百分比完成率（Percent Complete，PC）和计划进度百分比完成率（Planned Percent Complete，PPC）。

实际进度百分比完成率是指在截至日期为止实际完成的工作占全部工作量的百分比。根据题目描述，实际完成了其中 8.5 万元的工作，而总工作量为 10 万元，因此其实际进度百分比完成率 PC 为 85%。

计划进度百分比完成率是指在预算计划中规定的截止日期前计划完成的工作量占全部工作量的百分比。根据题目描述，该项目预计到 2015 年 10 月 10 日完成 10 万元的工作，因此其计划进度百分比完成率 PPC 为 100%。

3. 计算成本偏差（Cost Variance，CV）和进度偏差（Schedule Variance，SV）。

成本偏差是指实际成本与预算成本之间的差异。根据上述计算，BCWS = 10 万元、ACWP = 9 万元，因此成本偏差 CV = BCWS - ACWP = 1 万元。

进度偏差是指实际进度百分比完成率与计划进度百分比完成率之间的差异。根据上述计算，PC = 85%、PPC = 100%，因此进度偏差 SV = BCWS × (PC - PPC) = -1 万元。

4. 计算成本效能指数（Cost Performance Index，CPI）和进度效能指数（Schedule Performance Index，SPI）。

成本效能指数是指实际成本与挣值之间的比率。挣值（Earned Value，EV）是指在预算计划中规定的截止日期前按照计划完成的工作应该具有的价值。根据上述计算，PC = 85%，因此 EV = BCWS × PC = 8.5 万元。因此，成本效能指数 CPI = EV / ACWP = 0.9444。

进度效能指数是指实际进度百分比完成率与挣值百分比完成率之间的比率。根据上述计算，PPC = 100%，因此挣值百分比完成率为 PC = 85%。因此，进度效能指数 SPI = PC / PPC = 0.85。

**课后题 CH1\_1 项目管理知识体系包括哪5个过程？哪10个知识领域?**

项目管理知识体系包括以下5个过程：

1. 启动过程：确定项目目标，定义项目范围，明确项目可行性和风险。

2. 规划过程：制定详细的项目计划，包括时间、成本、资源等方面。

3. 执行过程：按照计划执行项目，组织资源，协调各方面工作。

4. 监控过程：监督和控制项目进展情况，调整计划，纠正偏差。

5. 收尾过程：完成项目交付物并进行验收，归档项目文档，总结经验教训。

项目管理知识体系包括以下10个知识领域：

1. 项目整合管理：负责协调和整合项目的各个方面工作，以确保项目成功完成。

2. 项目范围管理：负责确保项目的需求和目标得到满足，并设置控制机制以防止范围蔓延。

3. 项目时间管理：负责安排项目进度，确定关键路径和里程碑，以确保项目按时完成。

4. 项目成本管理：负责预算和管理项目成本，以确保项目在可接受的成本范围内完成。

5. 项目质量管理：规划和实施项目质量控制和保证，以确保项目交付的成果符合质量标准。

6. 项目人力资源管理：负责组织、管理和领导项目团队，确保团队成员能够高效地工作。

7. 项目沟通管理：规划和实施项目的沟通策略和计划，确保项目相关方之间的沟通顺畅。

8. 项目风险管理：识别、评估和应对项目中可能出现的各种风险，以确保项目成功完成。

9. 项目采购管理：负责规划和实施项目采购策略和计划，确定采购商品或服务的来源。

10. 项目干系人管理：负责管理和协调项目的所有利益相关者，包括客户、用户、股东等，以确保他们的需求得到满足。

**CH1\_2 软件项目管理一般分为哪几个阶段？个阶段主要任务是什么？**

软件项目管理通常分为以下几个主要阶段：

1. 需求分析阶段：在这个阶段，项目团队将与客户沟通并理解客户的需求和期望。主要任务是收集、分析、明确和记录客户需求，编写需求规格说明书，确定产品范围，并制定详细的项目计划。

2. 设计阶段：在这个阶段，项目团队将开发出满足需求规格说明书的系统设计。主要任务是定义系统架构设计，编写技术规格说明书，选择适当的开发工具和技术，以及定义详细的测试计划。

3. 实施阶段：在这个阶段，项目团队开始根据设计规格说明书和测试计划实现系统。主要任务包括编写代码、进行单元测试和集成测试，并在完成开发后发布版本。

4. 测试和验收阶段：在这个阶段，项目团队将测试和验证实现的系统是否符合需求规格说明书中的要求。主要任务是执行各种类型的测试（如系统测试、回归测试、性能测试等），修复错误，确保最终的交付物符合质量标准，并且符合客户的预期。

5. 交付和维护阶段：在这个阶段，项目团队将交付最终的产品，并进入后期维护阶段。主要任务是协助客户进行系统上线，提供必要的培训和支持，并持续跟踪系统性能和反馈，确保系统的稳定性和可用性。

**CH2 在项目招投标阶段，甲乙双方的任务分别是什么？**

甲方的任务：1. 确定项目需求和范围，编写招标文件，明确项目目标、任务和要求。2. 公开招标，发布招标公告，接收并审查投标者提交的投标文件。3. 评估投标文件，选择最具竞争力的中标方，并与其签订合同。4. 监督和管理整个项目实施过程，确保项目达到预期的质量、进度和成本要求。5. 对项目执行过程中可能发生的变更进行管理和控制，协调各方面利益相关者，保证项目顺利完成。

乙方的任务：1. 分析招标文件，了解项目需求和范围，确定是否满足自身能力和资源条件。

2. 准备投标文件，包括技术方案、商业报价等，并提交给甲方。3. 在投标文件中清晰地阐述自己的优势和特点，并展示出满足项目需求的能力和经验。4. 参加投标评选活动，向甲方提供必要的支持和解释，以增加自己的中标机会。5. 如果被选为中标方，需要全力配合甲方实现项目目标，确保项目顺利完成，并履行合同义务。

**CH3\_1 软件需求管理通常包括那些过程？**

1. 需求收集：在这个阶段，项目团队与客户沟通并理解客户的需求和期望。主要任务是收集、分析、明确和记录客户需求，编写需求规格说明书，确定产品范围，并制定详细的项目计划。

2. 需求分析：在这个阶段，项目团队将对需求进行详细分析，以便更好地理解需求并为后续工作做好准备。主要任务包括定义功能和非功能需求，建立用例模型和场景分析，梳理系统流程等。

3. 需求规划：在这个阶段，项目团队将制定详细的需求规划，以确保需求的完整性和一致性。主要任务包括识别和管理不同类型的需求、制定需求变更管理策略、建立需求跟踪矩阵等。

4. 需求确认：在这个阶段，项目团队确保需求规格说明书中所有需求都得到确认，并满足客户的预期。主要任务包括与客户确认需求、建立需求验收标准、评估需求的优先级等。

5. 需求控制：在这个阶段，项目团队通过应对变更和风险，控制需求的稳定性和一致性。主要任务包括创建变更请求、审批变更请求、修改需求规格说明书、跟踪变更请求状态等。

**CH3\_2 试写出任务分解的方法和步骤**

任务分解是将大型复杂项目分解为更小、更易管理的任务和活动的过程。以下是常用的任务分解方法和步骤：

1. 定义项目目标：首先需要明确项目的总体目标，以便确定需要完成的任务和活动。

2. 列举项目任务：将项目目标分解为具体的任务和活动，一般从高层次向低层次逐步细分。可以使用思维导图、流程图等工具来帮助列举。

3. 分析任务依赖关系：对每个任务进行分析，确定它们之间的依赖关系。这有助于确定任务执行的先后顺序和时间安排。

4. 制定任务清单：将所有任务和活动记录在任务清单中，并将它们组织成一个逻辑结构。

5. 确定任务优先级：根据项目目标和时间限制，确定每个任务的优先级。这有助于分配资源、优化执行顺序和保证项目进度。

6. 制定任务计划：根据任务清单和优先级，制定详细的任务计划。计划应包括任务的开始和结束时间、分派给谁以及完成所需的资源等信息。

7. 分配责任：根据任务计划，将任务分配给团队成员，并明确各自的责任和任务完成标准。

8. 监控任务进展：在整个任务分解的过程中，需要不断监控任务的进展情况，及时进行调整和协调，以确保项目按计划顺利完成。

**CH6\_1 请描述项目实施过程中可能出现的质量问题**

在项目实施过程中，可能会出现以下质量问题：

1. 没有明确定义的需求：如果项目的需求没有经过充分的研究和讨论，或者没有得到各方面的共识，那么在开发过程中就容易出现功能缺陷或不符合用户期望的情况。

2. 不完整的测试覆盖范围：测试用例设计不充分、测试场景和数据不全面等原因导致测试覆盖范围不足，这将影响软件系统的稳定性和可靠性。

3. 缺乏测试人员的支持：如果测试人员不参与项目的开发和需求讨论阶段，他们可能无法了解系统的设计和实现细节，从而无法有效地进行测试。

4. 时间不足：如果项目时间紧迫，开发团队可能会缩短测试时间来保证项目按时交付，这将增加软件错误被忽略的风险。

5. 没有有效的变更管理：在项目开发过程中，可能需要针对需求变化或技术限制进行修改。如果没有一个严格的变更管理流程，就会导致不必要的修改和冲突，从而影响软件的质量和稳定性。

6. 代码质量问题：如果开发人员的编码技巧和标准不高，会导致代码可读性差、容易出现bug或难以维护。

这些问题可能会导致软件质量下降、项目延期或者其他风险的发生。因此，在项目实施过程中，需要有明确的质量保证计划和相应的执行措施来确保软件质量。

**CH6——2 为什么确保软件过程的质量就能导致较高质量的软件产品？讨论QA的主要困难**

确保软件过程的质量是提高软件产品质量的关键。以下是一些原因：

1. 将错误尽早发现和解决：在软件开发周期中，早期发现问题和进行修复要比在后期进行更加容易和经济。如果在项目早期就开始着重质量，可以帮助团队及时检测和解决问题，从而减少软件缺陷数量和成本。

2. 提高软件可维护性：通过遵循良好的代码实践、文档编写标准以及使用版本控制等最佳实践，可以提高软件的可维护性。这有助于随着时间的推移，代码库不断增长和升级。

3. 改善客户满意度：如果一个软件产品能够按照客户需求且无错地完成工作，客户就会感到满意并且愿意继续使用该软件。

4. 降低风险：通过确保软件过程的质量，可以减少项目风险，这可能会导致额外的成本和时间消耗。

然而，确保软件过程质量并不是没有挑战的，以下是一些主要的困难：

1. 需求变化：随着软件项目的不断发展，需求和目标也可能会随之改变。这可能导致质量保证计划的重新评估和修订。

2. 时间压力：在开发时间紧迫的情况下，软件质量保证过程可能被削减或缩短，以确保项目按时完成。

3. 技术复杂性：某些软件项目非常复杂，可能需要特殊技能和经验的团队成员才能完成。

4. 人为因素：人为错误、疏忽或欺诈是最常见的软件质量问题之一。因此，需要对团队进行培训和考核，以确保所有成员都能够遵循最佳实践并尽可能减少这些错误的风险。

1. 缺乏自动化：如果没有合适的测试工具和流程来支持软件质量控制，将会使得软件测试变得更加繁琐和耗时。因此，自动化测试是一个重要的解决方案。

**CH7——1 写出配置管理的基本过程**

配置管理是软件工程中的关键过程，它有助于确保软件产品的质量，可靠性和稳定性。以下是配置管理的基本过程：

1. 配置标识：为软件产品建立唯一的和可检测的标识。这通常包括版本号、日期时间戳、构建号或其他标识信息，以便可以在不同的环境中对软件进行追踪和区分。

2. 变更控制：此过程确保所有变更都记录在案，并以统一的方式加以处理。每个变更都需要经过审查和批准，确保变更的必要性和合理性。审核人员必须记录和跟踪变更的状态并向利益相关者汇报进展情况。

3. 版本控制：此过程用于组织和跟踪软件代码的版本和修订历史。它包括对源代码、文档和其他重要文件的版本控制和存储。版本控制系统还提供了协作和共享开发资源的能力。

4. 构建与发布：此过程涉及生成可交付软件所需的文件，并将其打包到一个或多个文件中。这包括源代码编译、数据转换和其他处理步骤，以生成目标软件产品。构建后的输出文件必须得到验证并在发布前进行严格的测试。

5. 配置审计：此过程用于验证软件配置和文档是否正确，并检查所有变更记录是否准确、完整和一致，以及其他配置管理活动是否按照计划执行。审计人员还需要评估项目进度和质量，以确定是否存在潜在的风险或问题。

以上是基本的配置管理过程，这些步骤可以根据特定项目和组织需求而有所不同。它们通常需要配合使用，以确保软件产品的质量、可靠性和稳定性。

**CH7——2 简述配置管理在软件开发中的作用，并列举至少两种配置管理工具**

配置管理是软件开发中至关重要的步骤。其作用主要包括：

1. 确保软件产品的质量、稳定性和可靠性：通过为每个软件版本分配唯一的标识符、跟踪和记录变更历史，以及实现审计、版本控制和构建与发布流程等系列过程，配置管理有助于确保软件产品的质量、稳定性和可靠性。

2. 提高团队协作效率：配置管理工具可以帮助团队协调工作，确保各个成员都在同一版本上工作，并且能够轻松地共享代码和文档等资源文件，从而提高团队合作的效率。

3. 支持追溯和问题解决：当软件出现问题时，配置管理工具可以帮助开发者快速定位并修复问题。由于每个软件版本都经过标识并记录了变更历史，因此可以根据特定版本来追溯问题，并且追溯到具体的修改记录，以便更容易地解决问题。

**常见的配置管理工具有很多，以下是两种比较流行的工具：**

1. Git：Git 是一个非常流行的分布式版本控制系统，它支持多个开发者同时对同一代码库进行操作。它具有强大的版本控制和分支合并功能，可以帮助开发者更好地管理代码库，并且能够跟踪代码变化历史。

1. Jenkins：Jenkins 是一个流行的自动化构建工具，它可以帮助开发人员实现持续集成和持续交付（CI/CD）。通过使用 Jenkins，开发者可以定义一系列构建步骤，并在每次提交代码时自动运行这些步骤，从而确保软件产品始终保持最新状态，并且符合预期要求。

**CH8 项目组织结构通常由哪几种？各自优缺点有哪些？**

在软件项目中，常见的几种组织结构包括以下几种：

1. 功能型组织结构：在这种结构下，项目团队按照各自的职能领域进行划分，例如开发、测试、UI设计等。这种组织结构通常适用于大型公司或企业，具有高度的专业化和标准化，但会有协调和沟通方面的问题。

2. 项目型组织结构：在这种结构下，项目团队是由不同职能部门的专家组成，他们共同完成特定的项目。这种组织结构通常适用于需要快速响应市场需求的小型或中型公司。优点是团队成员之间的沟通和协作容易，但可能会因为职能部门之间的利益冲突而影响项目进展。

3. 矩阵型组织结构：在这种结构下，项目团队由各个职能部门的专家组成，并且每个人在多个项目中都会参与。这种组织结构可以最大限度地利用资源，提高生产力和效率。但是，它需要更复杂的管理，以确保团队成员之间的协作和沟通。

4. 混合型组织结构：这种结构是各种组织结构的混合体，以满足具体项目的需求。这种组织结构通常需要更复杂的管理和沟通，但可以更好地适应不同类型的项目和挑战。

不同的组织结构有其各自的优点和缺点，如下所示：

1. 功能型组织结构

优点：专业化水平高，能够提供高质量的产品。通过使用标准化流程和程序，可以减少错误和冲突的发生。

缺点：团队成员之间的沟通和协作较困难。缺乏灵活性，很难应对快速变化的市场需求。

2. 项目型组织结构

优点：团队成员之间的沟通和协作相对容易。可以快速响应市场需求，并且非常灵活。

缺点：职能部门之间可能会存在利益冲突。没有充分利用资源的潜力。

3. 矩阵型组织结构

优点：可以最大限度地利用资源，提高生产力和效率。团队成员之间的沟通和协作相对容易。

缺点：更复杂的管理要求，需要耗费更多的精力和时间。可能存在团队成员之间不同项目的冲突。

4. 混合型组织结构

优点：1 能够最大程度地适应不同类型的项目和挑战。2 可以提供更灵活的管理和沟通方式。

缺点：1 更复杂的管理要求，可能需要额外的资源和财力支持。2 团队成员之间需要有良好的协作和沟通能力。

**概念**

**项目管理：**PMBOK：运用各种相关技能、方法和工具，为满足或超越项目有关各方对项目的要求与期望，所开展的各种计划、组织和领导，控制方面的活动。